BAB II

KAJIAN TEORI

Kajian teori pada bab ini membahas tentang pengertian dan penjelasan yang berkaitan dengan optimasi, pemrograman linear, pemrograman nonlinear, *quadratic programming* dan algoritma genetika. Kajian teori tersebut akan digunakan untuk bab pembahasan selanjutnya.

A. Optimasi

Optimasi berasal dari bahasa inggris *optimization* yang memiliki arti memaksimalkan atau meminimumkan sebuah fungsi yang diberikan untuk beberapa macam kendala (Licker, 2003 : 170). Definisi lain yaitu menurut Rao (2009 : 1) optimasi dapat didefinisikan sebagai proses untuk menemukan kondisi yang memberikan nilai maksimum dan minimum dari suatu fungsi.

Optimasi berkaitan dengan pencarian solusi dari suatu permasalahan dengan kendala tertentu. Permasalahan tersebut dapat berupa pemrograman linear dan pemrograman nonlinear. Pada permasalahan sehari-hari seringkali diselesaikan dengan pemrograman non linear. Terdapat beberapa metode untuk menyelesaikan permasalahan pemrograman nonlinear, diantaranya adalah *Lagrange Multiplier*, pendekatan kondisi *Karush-Kuhn-Tucker*, *Quadratic Programming*, dan *Separable Programming*.

B. Fungsi Konveks dan Konkaf

Pada subbab ini akan didefinisikan mengenai beberapa fungsi sebagai berikut:

Definisi 2.1 (Varberg & Purcell, 2001 : 155)

Misalkan f terdefinisi pada interval I (terbuka, tertutup, atau tidak satupun) maka,

i. f *naik pada I jika untuk setiap pasang bilangan* x_1 *dan* x_2 *dalam I,*

$x_1 < x_2 \Rightarrow f(x_1) < f(x_2)$.

ii. f *turun pada I jika untuk setiap pasang bilangan* x_1 *dan* x_2 *dalam I,*

$x_1 > x_2 \Rightarrow f(x_1) > f(x_2)$

iii. f *monoton murni pada I jika f naik atau turun pada I.*

Definisi 2.2 (Varberg & Purcell, 2001 : 156)

Misalkan f terdiferensial untuk setiap $x \in R$, *maka f dikatakan cekung keatas atau konveks jika* $f'(x)$ *naik untuk setiap* $x \in R$ *dan f dikatakan cekung kebawah atau konkaf jika* $f'(x)$ *turun untuk setiap* $x \in R$.

Teorema 2.1 (Varberg & Purcell, 2001 : 157)

Misalkan f terdiferensial dua kali untuk setiap $x \in R$

i. *Jika* $f''(x) > 0$, *maka f cekung keatas atau konveks untuk setiap* $x \in R$,

ii. *Jika* $f''(x) < 0$, *maka f cekung kebawah atau konkaf untuk setiap* $x \in R$.

Turunan kedua dari fungsi *f* adalah turunan pertama dari f', sehingga dapat disimpulkan bahwa jika f'' positif maka f' naik dan jika f'' negatif maka f' turun.

Turunan parsial kedua dapat digunakan untuk menguji konveks atau konkafnya suatu fungsi *f* dengan banyak variabel. Uji konveksitas untuk fungsi satu variabel $f(x)$ yang memiliki turunan kedua untuk setiap *x* yang mungkin. Dengan demikian, menurut Hiller dan Lieberman (2008 : 473), fungsi $f(x)$ bersifat sebagai berikut:

1. Konveks jika dan hanya jika $\frac{\partial^2 f(x)}{\partial x^2} \geq 0$ untuk setiap nilai *x* yang mungkin,
2. Konveks ketat jika dan hanya jika $\frac{\partial^2 f(x)}{\partial x^2} > 0$ untuk setiap nilai *x* yang mungkin,
3. Konkaf jika dan hanya jika $\frac{\partial^2 f(x)}{\partial x^2} \leq 0$ untuk setiap nilai *x* yang mungkin,
4. Konkaf ketat jika dan hanya jika $\frac{\partial^2 f(x)}{\partial x^2} < 0$ untuk setiap nilai *x* yang mungkin.

Jika terdapat dua variabel (x_1, x_2) maka uji konveksitas dapat dilihat pada tabel 2.1 berikut:

Tabel 2.1. Fungsi Konveks dan Konkaf dengan Dua Variabel

Kuantitas	Konveks	Konveks Ketat	Konkaf	Konkaf Ketat
$\frac{\partial^2 f(x_1,x_2)}{\partial {x_1}^2}\frac{\partial^2 f(x_1,x_2)}{\partial {x_2}^2} - \left[\frac{\partial^2 f(x_1,x_2)}{\partial {x_1}^2 \partial {x_2}^2}\right]$	≥ 0	> 0	≤ 0	< 0
$\frac{\partial^2 f(x_1,x_2)}{\partial {x_1}^2}$	≥ 0	> 0	≤ 0	< 0
$\frac{\partial^2 f(x_1,x_2)}{\partial {x_2}^2}$	≥ 0	> 0	≤ 0	< 0

C. Pemrograman Linear

Pemrograman Linear adalah metode optimasi untuk menemukan nilai optimum dan fungsi tujuan linear pada kondisi pembatasan-pembatasan tertentu (Ruminta, 2009 : 327). Pemrograman linear adalah salah satu teknik *Operations*

Research yang paling banyak digunakan. Teknik ini menjadi dasar pengembangan teknik *Operations Rersearch* yang lain seperti *Goal Programming, Binary Integer Programming* atau *Zero-One Programming.* Teknik ini bisa digunakan untuk menyelesaikan teknik *Operations Research* lain, seperti *Transportation, Assignment, Crash Time and Crash Cost program* pada *Critical Path Method, Equilibrium condition* pada *Markov Analysis*, *Dual Programming* pada *Game Theory*, dan *Network Analysis* seperti *Transhipment, Shortest Route, Minimum Spanning Tree*, dan *Maximal Flow*.

Model pemrograman linear mempunyai tiga unsur utama yaitu (Siswanto, 2006 : 25-26):

1. Variabel Keputusan

   Variabel Keputusan adalah variabel yang akan mempengaruhi nilai tujuan yang hendak dicapai. Pada proses pembentukan suatu model, menentukan variabel keputusan harus dilakukan terlebih dahulu sebelum menentukan fungsi tujuan dan kendala-kendalanya.

2. Fungsi Tujuan

   Fungsi Tujuan adalah tujuan yang hendak dicapai diwujudkan ke dalam sebuah fungsi matematika linear. Selanjutnya fungsi-fungsi tersebut dimaksimumkan atau diminimumkan terhadap kendala-kendala yang ada. Terdapat dua kemungkinan fungsi tujuan yaitu,

   1) Maksimumkan $f(X)$

   2) Minimumkan $f(X)$

3. Fungsi Kendala

   Fungsi Kendala adalah suatu pembatas terhadap variabel-variabel keputusan yang dibuat. Kendala tersebut harus dituangkan ke dalam fungsi matematika linear. Dapat disimpulkan bahwa program linear adalah sebuah metode matematis yang berkarakteristik linear untuk menemukan suatu penyelesaian optimal dengan cara memaksimumkan atau meminimumkan fungsi tujuan terhadap satu susunan kendala.

Beberapa asumsi penggunaan pemrograman linear pada kehidupan nyata yaitu (Ruminta, 2009 : 329):

1. Proporsionalitas

   Kontribusi masing-masing variabel keputusan terhadap fungsi tujuan dan pembatasan-pembatasan adalah proporsional langsung terhadap nilai variabel keputusan.

2. Aditivitas

   Kontribusi terhadap fungsi tujuan dan pembatasan-pembatasan untuk beberapa variabel keputusan yang lain sehingga kontribusi masing-masing variabel keputusan dapat digabungkan / ditambahkan menjadi kontribusi total.

3. Divisibilitas

   Variabel keputusan adalah kontinu sehingga dapat diambil nilai fraksionalnya.

4. Deterministik

   Semua parameter (fungsi tujuan, pembatasan-pembatasan, seluruh koefisien) diketahui dengan pasti dan tetap tidak berubah selama dilakukan kajian dan analisis.

Menurut Ruminta (2009 : 330), persoalan program linear adalah persoalan optimasi yang memenuhi ketentuan berikut:

1. Fungsi tujuan merupakan fungsi linear dari variabel keputusan.
2. Nilai variabel keputusan harus memenuhi pembatasan-pembatasan. Setiap pembatasan harus berbentuk persamaan atau ketidaksamaan linear.
3. Setiap variabel keputusan harus dibatasi yaitu non negatif.

Definisi 2.3 Fungsi Linear (Winston, 2004 : 52)

Fungsi $(x_1, x_2, \dots, x_n)$ *merupakan fungsi linear jika dan hanya jika fungsi* f *dapat dituliskan* $f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$, *dengan* $c_1, c_2, \dots, c_n$ *merupakan konstanta.*

Contoh 2.1

Fungsi berikut merupakan fungsi linear:

$f(x_1, x_2) = 3x_1 + x_2$

Definisi 2.4 Fungsi Pertidaksamaan Linear (Winston, 2004 : 52)

Untuk sebarang fungsi linear $f(x_1, x_2, \dots, x_n)$ *dan sebarang bilangan* $b \epsilon R$, *pertidaksamaan* $f(x_1, x_2, \dots, x_n) \leq b$ *dan* $f(x_1, x_2, \dots, x_n) \geq b$ *merupakan fungsi pertidaksamaan linear.*

Contoh 2.2

Fungsi berikut merupakan fungsi pertidaksamaan linear:

$5x_1 + 2x_2 \leq 2$

Ada beberapa persyaratan penting dalam merumuskan persoalan pemrograman linear yaitu (Ruminta, 2009 : 331):

1. Ada beberapa kuantitas yang memungkinkan dioptimasi untuk digunakan sebagai tujuan.
2. Ada variabel-variabel yang dapat dibuat variabel keputusan.
3. Ada pembatasan kemampuan dalam mencapai tujuan.
4. Ada langkah-langkah alternatif pemecahan yang dapat dipilih.
5. Tujuan dan pembatasan-pembatasan harus dapat diekspresikan dalam persamaan atau ketidaksamaan linear.

Definisi 2.5 Pemrograman Linear (Susanta, 1994 : 6)

Diberikan fungsi $f(x_1, x_2, \dots, x_n)$ *merupakan fungsi linear,* x_j *merupakan variabel keputusan ke-j,* c_j *dan* a_{ij} *merupakan konstanta-konstanta yang diketahui,* b_m *merupakan nilai ruas kanan dari persamaan kendala ke-m yang menunjukkan nilai syarat kendala tersebut, umtuk setiap* $i = 1, 2, \dots, m$ *(indeks untuk jumlah variabel kendala dan* $j = 1, 2, \dots, n$ *(indeks untuk jumlah variabel keputusan.*

Secara umum, model pemrograman linear dirumuskan sebagai berikut:

Memaksimumkan / Meminimumkan : $f = C^T X$ (2.1)

dengan kendala : $AX(\leq, =, \geq)B,$ dan $X \geq 0$ (2.2) dalam hal ini, $C^T = [c_1 \quad c_2 \quad \dots \quad c_n],$ (2.3)

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \tag{2.4}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \tag{2.5}$$

dan $B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$ (2.6)

Matriks X merupakan matriks satu kolom dari variabel-variabel yang dicari, dan C^T adalah matriks satu baris untuk setiap koefisien ongkos (c_j). Matriks A merupakan matriks koefisien persamaan kendala, dan B adalah matriks satu kolom dari ruas kanan persamaan kendala. (Bronson & Naadimuthu, 1997 : 20)

Jika (2.1) dan (2.2) dituliskan dalam bentuk matriks menjadi:

Memaksimumkan / Meminimumkan : $f = [c_1 \quad c_2 \quad \dots \quad c_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},$

dengan kendala :

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} (\leq, =, \geq) \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \text{ dan } \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \geq 0.$$

Jika bentuk perkalian matriks tersebut diuraikan menjadi penjumlahan aljabar akan menjadi:

Memaksimumkan / Meminimumkan

$$f = c_1x_1 + c_2x_2 + \cdots + c_nx_n = \sum_{j=1}^{n} c_jx_j \quad (2.7)$$

dengan kendala:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n\ (\leq, =, \geq) b_1 \quad (2.8a)$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n (\leq, =, \geq) b_2 \quad (2.8b)$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n (\leq, =, \geq) b_n \quad (2.8c)$$

$$x_1, x_2, \dots, x_n \geq 0 \quad (2.8d)$$

atau jika ditulis ulang, maka bentuk fungsi kendala (2.8a) - (2.8d) menjadi:

$g_i(x_1, x_2, \dots, x_n) = \sum_{j=1}^{n} a_{ij}x_j\ (\leq, =, \geq) b_i$ dengan $i = 1, 2, \dots, m$ (2.9a)

$x_j \geq 0$, dengan $j = 1, 2, \dots, n$ (2.9b)

D. Pemrograman Nonlinear

Pemrograman nonlinear adalah suatu teknik dalam masalah optimasi yang mempunyai fungsi tujuan nonlinear dan fungsi kendala berbentuk nonlinear atau linear. (Bazaraa et al, 2006 : 1).

Bentuk umum dari pemrograman nonlinear adalah menemukan nilai $X = (x_1, x_2, \dots, x_n)$, (Hiller & Lieberman, 2001 : 654) sehingga

Max / Min $f(X)$,

dimana $f(X)$ berupa fungsi non linear, (2.10a)

kendala $g_i(X)(\leq, =, \geq) b_i$, untuk setiap $i = 1, 2, \dots, m$ (2.10b)

dan $x_j \geq 0, i = 1, 2, \dots, n$ (2.10c)

Fungsi kendala $g_i\,(X)$ dapat berupa fungsi nonlinear maupun fungsi linear. Selain itu, $f(X)$ dan fungsi $g_i\,(X)$ adalah fungsi-fungsi dengan n variabel.

E. *Quadratic Programming*

Quadratic Programming adalah pendekatan penyelesaian permasalahan optimasi nonlinear dimana kendalanya berupa fungsi linear dan fungsi tujuannya merupakan kuadrat dari variabel keputusan ataupun perkalian dari dua variabel keputusan (Hiller & Lieberman, 2001 : 665). Bentuk umum dari masalah *quadratic programming* menurut Peressini et al (1988 : 258) yaitu:

Meminimumkan $f(X) = C^T X + \frac{1}{2} X^T Q X + d$ (2.11a)

Dengan kendala $AX \leq B, X \geq 0$ (2.11b)

Konsep matriks C^T, X, A, dan B sama seperti penjelasan pada (2.3) – (2.6). adapun d merupakan suatu konnstanta, sedangkan Q merupakan matriks simetris yang tersusun dari nilai q_{ij}, dimana q_{ij} merupakan hasil dari turunan parsial kedua terhadap x_i dan x_j dari fungsi tujuan. Matriks Q merupakan matriks simetris, sehingga nilai $q_{ij} = q_{ji}$. Bentuk (2.11a) dapat ditransformasikan menjadi berikut:

$$f(X) = C^T X + \frac{1}{2} X^T Q X + d = \sum_{j=1}^{n} c_j x_j + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j + d \qquad (2.12)$$

Jika Q adalah matriks definit positif, $f(X)$ merupakan fungsi konveks ketat, dan fungsi kendala merupakan fungsi konveks, maka setiap nilai minimum dari masalah tersebut merupakan minimum global (Rao, 1984 : 231).

F. Kondisi Karush Khun-Tucker

Model Karush Khun – Tucker dapat digunakan untuk menyelesaikan suatu fungsi linear maupun non linear. Pada metode ini, program yang diselesaikan merupakan program yang memiliki kendala pertidaksamaan. Metode Karush Khun – Tucker merupakan pengembangan dari penyelesaian model nonlinear berkendala persamaan yang dikerjakan dengan mencari titik-titik stasionernya, yaitu titik yang berpotensi menjadi titik optimal.

Terdapat beberapa syarat Karush Khun – Tucker untuk masalah optimasi berkendala. Syarat tersebut dirumuskan oleh Karush dan Khun – Tucker. Berikut adalah teorema yang menjelasakan tentang syarat Karush Khun – Tucker untuk masalah maksimum dan minimum.

Teorema 2.2. Syarat Karush Khun-Tucker masalah maksimum (Winston, 2003 : 676)

Misalkan $f(X)$ *dan* $g_i(X)$ *merupakan suatu masalah berpola memaksimumkan. Jika* $X = (x_1, x_2, \dots, x_n)$ *merupakan suatu solusi optimal untuk* $f(X)$ *dan* $g_i(X)$*, maka* $X = (x_1, x_2, \dots, x_n)$ *merupakan fungsi nonlinear dan terdapat pengali* $\lambda_1, \lambda_2, \dots, \lambda_m$ *serta variabel slack* $s_1, s_2, \dots, s_n$ *sehingga memenuhi*

1. $\frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j} + s_j = 0,$ $\quad untuk\ j = 1, 2, \dots, n$
2. $\lambda_i [b_i - g_i(X)] = 0,$ $\quad untuk\ i = 1, 2, \dots, m$
3. $(\frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j}) x_j = 0,$ $\quad untuk\ j = 1, 2, \dots, n$
4. $\lambda_i \geq 0,$ $\quad untuk\ i = 1, 2, \dots, m$

5. $s_j \geq 0,$ $\quad$ $untuk\ j = 1, 2, \dots, n$

Teorema 2.3. Syarat Karush Khun-Tucker masalah minimum (Winston, 2003 : 676)

Misalkan $f(X)$ *dan* $g_i(X)$ *merupakan suatu masalah berpola meminimumkan. Jika* $X = (x_1, x_2, \dots, x_n)$ *merupakan suatu solusi optimal untuk* $f(X)$ *dan* $g_i(X)$*, maka* $X = (x_1, x_2, \dots, x_n)$ *merupakan fungsi nonlinear dan terdapat pengali* $\lambda_1, \lambda_2, \dots, \lambda_m$ *serta variabel surplus* $e_1, e_2, \dots, e_n$ *sehingga memenuhi*

1. $\frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j} - e_j = 0,$ $\quad$ $untuk\ j = 1, 2, \dots, n$
2. $\lambda_i [b_i - g_i(X)] = 0,$ $\quad$ $untuk\ i = 1, 2, \dots, m$
3. $\left(\frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j}\right) x_j = 0,$ $\quad$ $untuk\ j = 1, 2, \dots, n$
4. $\lambda_i \geq 0,$ $\quad$ $untuk\ i = 1, 2, \dots, m$
5. $e_j \geq 0,$ $\quad$ $untuk\ j = 1, 2, \dots, n$

Pada syarat kedua dari Teorema 2.2 dan Teorema 2.3 berakibat $g_i(X) - b_i \leq 0$. Hal ini dapat dilihat pada saat $\lambda_i = 0$, sehingga $[b_i - g_i(X)] \neq 0$. Berdasarkan bentuk umum fungsi kendala, maka $[b_i - g_i(X)] \geq 0$ yaitu $g_i(X) \leq b_i$.

Permasalahan pada *quadratic programming* dapat diselesaikan dengan menggunakan persyaratan Khun-Tucker seperti yang tertera pada Teorema 2.2 dan Teorema 2.3. Selain itu, dalam *quadratic programming* juga terdapat kondisi *complementary slackness* khusus.

Secara umum, kondisi *complementary slackness* pada *quadratic programming* dapat dinyatakan dalam Sifat 2.1 sebagai berikut:

Sifat 2.1. *Complementary slackness* pada *quadratic programming* (Winston, 2003 : 687)

1. e_j dan s_j pada kondisi Khun-Tucker dan x_j tidak dapat kedua-duanya bernilai positif.
2. Variabel *surplus* (*excess*) ataupun *slack* untuk kendala ke-*i* dan λ_i tidak dapat kedua-duanya bernilai positif.

Bukti Sifat 2.1:

1. Diperhatikan Syarat 1) dan 3) pada Teorema 2.2, yaitu:

   Syarat 1) yaitu : $\frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_1 \frac{\partial g_i}{\partial x_j} + s_j = 0$, sehingga

   $\frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_1 \frac{\partial g_i}{\partial x_j} = -s_j$ disubtitusikan ke Syarat 3)

   $$\left(\frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_1 \frac{\partial g_i}{\partial x_j}\right) x_j = 0$$

   $s_j x_j = 0.$

   Jika $s_j = 0$ maka $x_j \neq 0$, yaitu $x_j > 0$.

   Jika $x_j = 0$ maka $s_j \neq 0$, yaitu $s_j > 0$ atau $s_j < 0$. Berdasarkan Syarat 5) maka $s_j > 0$.

   Hal ini berlaku juga untuk Teorema 2.3, sehingga terbukti bahwa e_j dan s_j pada kondisi Khun-Tucker dan x_j tidak dapat kedua-duanya bernilai positif.

2. Diperhatikan Syarat 2) yaitu $\lambda_i[b_i - g_i(X)] = 0$

   Pada fungsi kendala $g_i(X) \leq b_i$ maka bentuk kanonik kendala tersebut yaitu $g_i(X) + s_i' = b_i$, sehingga syarat 2) menjadi:

   $\lambda_i s_i' = 0$

   Jika $\lambda_i = 0$ maka $s_i' \neq 0$, yaitu $s_i' > 0$.

   Jika $s_i' = 0$ maka $\lambda_i \neq 0$, yaitu $\lambda_i > 0$ atau $\lambda_i < 0$. Berdasarkan Syarat 5) maka $\lambda_i > 0$.

   Pada fungsi kendala $g_i(X) \geq b_i$ dapat diubah menjadi $g_i(X) - e_i' = b_i$.

Melalui cara yang sama maka didapat pula $\lambda_i e_i' = 0$, sehingga terbukti bahwa variabel *surplus* (*excess*) ataupun *slack* untuk kendala ke-*i* dan λ_i tidak dapat kedua-duanya bernilai positif.

G. Algoritma Genetika

Algoritma Genetika merupakan suatu metode algoritma pencarian berdasarkan pada mekanisme seleksi alam dan genetik alam (Kusumadewi, 2003 : 87). Algoritma Genetika terinspirasi oleh proses biologi dari teori evolusi Darwin, sehingga banyak istilah dan konsep biologi yang digunakan dalam Algoritma Genetika (Chambers, 2000 : 13).

Algoritma Genetika muncul dari teori-teori dalam buku biologi, sehingga banyak istilah dan konsep biologi yang digunakan. Proses-proses yang terjadi dalam Algoritma Genetika juga tidak jauh berbeda dengan apa yang terjadi pada evolusi biologi. Sekumpulan individu yang sama, yang disebut spesies, hidup, berproduksi dan mati dalam satu area yang disebut populasi. Jika anggota-anggota populasi (individu) terpisah, misalnya karena terjadi banjir atau gempa, maka individu-individu tersebut akan membentuk beberapa populasi yang terpisah. Setelah beberapa waktu akan terjadi proses pembentukan spesies baru atau dikenal dengan istilah speciation dan juga terjadi perubahan hereditas (*heredity*) secara bertahap yang membentuk ciri-ciri baru pada spesies tersebut. Sebagai contoh, spesies pemangsa mengalami perubahan bertahap sehingga memiliki gigi taring yang lebih panjang dan tajam (Suyanto, 2005 : 1).

Algoritma Genetika merupakan teknik pencarian yang didasarkan atas mekanisme seleksi genetik natural. Algoritma genetika berbeda dengan teknik pencarian konvensional. Algoritma genetika dimulai dari himpunan solusi yang pada umumnya dihasilkan secara acak. Himpunan ini disebut populasi, sedangkan setiap individu dalam populasi disebut kromosom (merupakan representasi dari solusi) dan yang menempati kromosom disebut gen dan nilainya dapat berupa bilangan numerik, bilangan biner, simbol ataupun sebuah karakter dari permasalahan yang ingin diselesaikan (Gen & Cheng, 2000 : 1).

Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness* (kebugaran). Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas dari kromosom dalam populasi tersebut (Zukhri, 2014 : 23). Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*).

Selain operator penyilangan, suatu kromosom dapat pula dimodifikasi dengan menggunakan operator mutasi (*mutation*) dengan harapan akan menghasilkan kromosom baru dengan tingkat fitness lebih tinggi sebagai generasi baru atau keturunan (*offspring*) berikutnya. Setelah beberapa generasi maka algoritma genetika akan konvergen pada kromosom terbaik, yang diharapkan merupakan solusi optimal (Goldberg, 1989 : 71).

Menurut Gen dan Cheng ada tiga kelebihan dari Algorima Genetika dalam proses pencarian nilai optimal (Zukhri, 2014 : 11), yaitu:

1. Algoritma Genetika hanya memerlukan sedikit perhitungan matematis yang berhubungan dengan masalah yang ingin diselesaikan.
2. Operasi evolusi dari Algoritma Genetika sangat efektif untuk mengobservasi posisi global secara acak.
3. Algoritma genetika memiliki fleksibilitas untuk diimplementasikan secara efisien pada problematika tertentu.

Hal-hal yang terdapat dalam Algoritma Genetika adalah sebagai berikut (Weise, 2009):

a. Gen (*Genotype*) adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom.
b. *Allele* yaitu nilai dari sebuah gen, dapat berupa bilangan biner, float, integer, karakter, dan kombinatorial.
c. Kromosom adalah gabungan gen-gen yang membentuk nilai tertentu.

d. Individu merupakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
e. Populasi merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi. Populasi terdiri dari sekumpulan kromosom.
f. Induk adalah kromosom yang akan dikenai operasi genetik pindah silang.
g. Pindah silang merupakan operasi genetik yang mewakili proses perkembangbiakan antar individu.
h. *Offspring* adalah kromosom yang merupakan hasil dari operasi genetik pindah silang yang dikenal keturunan atau sebagai anak.
i. Mutasi merupakan operasi genetik yang mewakili proses mutasi dalam perjalanan hidup individu. Mutasi berperan menghasilkan perubahan acak dalam populasi, yang berguna untuk menambah variasi dari kromosom-kromosom dalam sebuah populasi.
j. Proses seleksi merupakan proses yang mewakili proses seleksi alam (*natural selection*) dari teori Darwin. Proses ini dilakukan untuk menentukan induk dari operasi genetik pindah silang yang akan dilakukan untuk menghasilkan keturunan (*offspring*).
k. Nilai *fitness* merupakan penilaian yang menentukan bagus tidaknya sebuah kromosom.
l. Fungsi Evaluasi adalah fungsi yang digunakan untuk menentukan nilai fitness. Fungsi evaluasi ini merupakan sekumpulan kriteria-kriteria tertentu dari permasalahan yang ingin diselesaikan.

m. Generasi merupakan satuan dari populasi setelah mengalami operasi-operasi genetika, berkembang biak, dan menghasilkan keturunan. Pada akhir dari setiap generasi, untuk menjaga jumlah kromosom dalam populasi tetap konstan, kromosom-kromosom yang mempunyai nilai fitness yang rendah dan memiliki peringkat dibawah nilai minimal akan dihapus dari populasi.

Algoritma Genetika sangat tepat jika digunakan untuk menyelesaikan masalah optimasi yang kompleks dan sukar diselesaikan dengan menggunakan metode konvensional (Supriyanto, 2010 : 4). Sebagaimana halnya dengan proses evolusi di alam, suatu algoritma genetikayang sederhana umumnya terdiri dari tiga operasi, yaitu: operasi seleksi, operasi *crossover* (persilangan), dan operasi mutasi.

Struktur umum dari suatu Algoritma Genetika terdiri dari langkah-langkah:

a. Membangkitkan Populasi Awal

Diawali dengan membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada (Kusumadewi, 2003 : 282).

Proses pembangkitan populasi awal diawali dari pengkodean gen dari kromosom. Satu gen biasanya merepresentasikan satu variabel. Gen dapat diwakili dalam bentuk bilangan *real*, bit, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika. Teknik pengkodean ini tergantung pada pemecahan masalah yang dihadapi seperti bilangan acak, pendekatan tertentu, dan permutasi gen.

b. Seleksi

Seleksi merupakan pemilihan dua buah kromosom untuk dijadikan sebagai induk yang dilakukan secara proporsional sesuai dengan nilai *fitness*-nya (Michalewics, 1996 : 75). Seleksi digunakan untuk memilih individu-individu mana saja yang akan dipilih untuk proses crossover dan mutasi. Masing-masing individu yang diseleksi akan diberikan probabilitas reproduksi tergantung dari nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya. Untuk itu dapat digunakan rumus sebagai berikut :

$$fitness = \frac{1}{1+nilai\ fungsi\ objektif} \qquad (2.13)$$

Fungsi objektif perlu ditambah 1 untuk menghindari kesalahan yang diakibatkan pembagian oleh 0. Semakin tinggi nilai *fitness* suatu individu semakin besar kemungkinannya untuk dipilih.

Masing-masing individu dalam wadah seleksi akan menerima probabilitas yang tergantung pada nilai *fitness*-nya. Selanjutnya akan dicari nilai probabilitasnya dengan rumus sebagai berikut:

$$P(i) = \frac{fitness\ (i)}{total\ fitness} \qquad (2.14)$$

Cara untuk mencari nilai kumulatif probabilitas adalah sebagai berikut:

$C(i) = \sum(fitness\ (i))$ (2.15)

Setelah didapat nilai probabilitasnya selanjutnya dihitung komulatif probabilitasnya. Proses seleksi menggunakan *roulete-wheel* dilakukan setelah didapatkan nilai komulatif probabilitas. Prosesnya adalah dengan membangkitkan bilangan acak R dalam *range* 0 – 1.

Jika $C(k-1) < R < C(k)$ maka pilih kromosom ke-*k* sebagai induk. (2.16)

c. *Crossover*

Pindah silang (*Crossover*) adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. *Crossover* menghasilkan keturunan baru dalam ruang pencarian yang siap diuji. Operasi ini tidak selalu dilakukan pada setiap individu yang ada. Individu dipilih secara acak untuk dilakukan *crossover* dengan parameter *crossover probability* (ρ_c) antara 0,6 – 0,95. Jika *crossover* tidak dilakukan maka nilai dari induk akan diturunkan kepada keturunan (Michalewicz, 1996 : 78).

Prinsip dari *crossover* adalah melakukan operasi genetika (pertukaran, aritmatika) pada gen-gen yang bersesuaian dari dua induk untuk menghasilkan indivadu baru. *Crossover* dilakukan pada setiap individu dengan probabilitas *crossover* yang telah ditentukan.

d. Mutasi

Mutasi merupakan proses untuk mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi mutasi yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatangdengan *fitness* yang lebih baik, dan lama kelamaan menuju solusi optimum yang diinginkan. Akan tetapi, untuk mencapai hal ini penekanan selektif juga memegang peranan yang penting. Jika dalam proses pemilihan kromosom-kromosom cenderung terus pada kromosom yang memiliki *fitness* yang tinggi saja, konvergensi prematur akan sangat mudah terjadi (Murniati, 2009 : 24).

Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh parameter *mutation probability* (ρ_m). Pada umumnya nilainya adalah $\frac{1}{n}$ (Supriyanto, 2010 : 10). Proses mutasi dilakukan dengan cara mengganti satu gen yang terpilih secara acak dengan suatu nilai baru yang didapat secara acak. Langkah pertama adalah menghitung panjang total gen yang ada dalam satu populasi.

$$panjang\ total\ gen = jumlah\ gen\ \times jumlah\ populasi \tag{2.17}$$

Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi jika peluang mutasi terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya dan algoritme kehilangan kemampuan untuk belajar dan melakukan pencarian.

e. Evaluasi Solusi

Langkah ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* dari setiap kromosom hingga kriteria menjadi terpenuhi. Namun karena seleksi dilakukan secara acak maka diperlukan langkah untuk menjaga agar individu bernilai *fitness* terbaik tidak hilang selama proses evolusi. Proses ini dikenal dengan nama *etilism* (Kusumadewi, 2003 : 112). Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah sebelumnya tetapi tetap menyertakan individu yang disimpan dalam proses *etilism* sehingga hasil perhitungan dapat konvergen. Beberapa kriteria berhenti menurut Sukmawan (2003 : 25) adalah sebagai berikut:

1. Berhenti pada generasi tertentu.
2. Berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi / terendah yang tidak berubah.
3. Berhenti bila dalam *n* generasi berikutnya tidak diperoleh nilai *fitness* yang lebih tinggi / rendah.

Pada umumnya proses Algoritma Genetika untuk mendapatkan hasil optimal membutuhkan proses pengulangan yang cukup panjang.

Selanjutnya, diberikan contoh pennyelesaian pemrograman nonlinier menggunakan metode *quadratic programming* dengan algoritma genetika. Berikut gambaran umum penyelesaian pemrograman nonlinear menggunakan metode *quadratic programming* dengan algoritma genetika.

Contoh 2.1

Meminimumkan fungsi :

$$f(x) = 2{x_1}^2 + 2{x_2}^2 + {x_3}^2 - 4x_1 - 6x_2 + 2x_3 \quad (2.16)$$

dengan kendala :

$$x_1 \geq 3 \quad (2.17a)$$

$$x_2 \geq 8 \quad (2.17b)$$

$x_3 \geq 2$ (2.17c)

$x_1, x_2,\ x_3\ \geq 0$ (2.17d)

Penyelesaian:

a. Pembentukan model nonlinear menggunakan *quadratic programming*

Persamaan (2.16) dan (2.17) perlu diidentifikasi menjadi bentuk umum *quadratic programming* yang tertera pada persamaan (2.10) dan (2.11) sebagai berikut:

Persamaan (2.16) dapat ditentukan :

Matriks C^T merupakan matriks baris koefisien-koefisien dari x sehingga:

$$C^T = \begin{bmatrix} -4 & -6 & 2 \end{bmatrix},$$

Matriks X adalah matriks kolom untuk variabel-variabel keputusan, sehingga $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$.

Matriks Q merupakan matriks simetris dengan q_{11} merupakan turunan kedua dari x_1, q_{22} merupakan turunan kedua dari x_2, dan q_{33} merupakan turunan kedua dari x_3, sehingga matriks Q:

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$f(x_1, x_2, x_3) = 2{x_1}^2 + 2{x_2}^2 + {x_3}^2 - 4x_1 - 6x_2 + 2x_4$$

$$f(X) = C^T X + \frac{1}{2} X^T Q X + d$$

$$= \begin{bmatrix} -4 & -6 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 0$$

Bentuk Kendala (2.17) kemudian disesuaikan dengan persamaan (2.11) menjadi:

$$x_1 \geq 3 \quad \Leftrightarrow \quad -x_1 \leq -3$$

$$x_2 \geq 8 \quad \Leftrightarrow \quad -x_2 \leq -8$$

$$x_1 \geq 2 \quad \Leftrightarrow \quad -x_3 \leq -2$$

$$x_1, x_2,\ x_3\ \geq 0$$

Tampak bahwa Persamaan (2.16) dan (2.17) memenuhi bentuk persamaan (2.10) dan (2.11). Selanjutnya, akan dilihat apakah persamaan (2.16) dan (2.17) menghasilkan minimum global jika diselesaikan menggunakan metode *quadratic programming* dengan algoritma genetika, yaitu dengan melihat turunan parsialnya. Diperoleh turunan parsial kedua dari persamaan (2.16) adalah sebagai berikut :

$$\frac{\partial^2 f(x)}{\partial {x_1}^2} = 4 > 0$$

$$\frac{\partial^2 f(x)}{\partial {x_2}^2} = 4 > 0$$

$$\frac{\partial^2 f(x)}{\partial {x_3}^2} = 2 > 0$$

dan turunan pertama dari Persamaan (2.17) adalah sebagai berikut:

$$\frac{\partial g_1(x)}{\partial x_1} = 1 > 0$$

$$\frac{\partial g_2(x)}{\partial x_2} = 1 > 0$$

$$\frac{\partial g_3(x)}{\partial x_3} = 1 > 0$$

Terlihat bahwa $\frac{\partial^2 f(x)}{\partial x^2} > 0$, maka berdasarkan Tabel 2.1 fungsi $f(x)$ merupakan fungsi koveks ketat, sedangkan $g'(x) = \frac{g(x)}{\partial x} > 0$, maka menurut Definisi 2.2 merupakan fungsi konveks. Karena $f(x)$ merupakan fungsi konveks dan $g(x)$ meriupakan fungsi konveks maka minimum lokal yang dihasilkan dari permasalahan tersebut merupakan minimum global (Rao, 1984 : 231). Sehingga, persamaan (2.16) cukup diselesaikan dengan *quadratic programming* menggunakan metode *Khun-Tucker* untuk mencari nilai minimum global (Rao, 1984 : 98).

Langkah-langkah penyelesaian dengan quadratic programming adalah sebagai berikut:

1. Menentukan kondisi *Khun Tucker*

   Berdasarkan Teorema 2.3 maka persamaan (2.16) dapat ditentukan syarat *Khun Tucker*-nya yaitu:

   1) $4x_1 - 4 - \lambda_1 - e_1 = 0$ (2.18a)

      $4x_2 - 6 - \lambda_2 - e_2 = 0$ (2.18b)

      $2x_3 + 2 - \lambda_3 - e_3 = 0$ (2.18c)

   2) $\lambda_1[-3 - (-x_1)] = 0$ (2.19a)

      $\lambda_2[-8 - (-x_2)] = 0$ (2.19b)

      $\lambda_3[-2 - (-x_3)] = 0$ (2.19c)

3) $(4\, x_1 - 4 - \lambda_1)x_1 = 0$ (2.20a)

$(4\, x_2 - 6 - \lambda_2)x_2 = 0$ (2.20b)

$(2\, x_3 + 2 - \lambda_3)x_3 = 0$ (2.20c)

4) $\lambda_1, \lambda_2, \lambda_3 \geq 0$ (2.21)

5) $e_1, e_2, e_3 \geq 0$ (2.22)

Sebagai akibat dari (2.19) maka:

$-x_1 - (-3) \leq 0$ (2.23a)

$-x_2 - (-8) \leq 0$ (2.23b)

$-x_3 - (-2) \leq 0$ (2.23c)

Bentuk (2.23) dapat dijadikan bentuk kanonik sehingga menjadi:

$x_1 - e_1' = 3$ (2.24a)

$x_2 - e_2' = 8$ (2.24b)

$x_3 - e_3' = 2$ (2.24c)

Setelah mengidentifikasi syarat *Khun Tucker*, maka kondisi Khun Tucker untuk persamaan (2.16) dan (2.17) yaitu:

$4x_1 - 4 - \lambda_1 - e_1 = 0$ (2.18a)

$4x_2 - 6 - \lambda_2 - e_2 = 0$ (2.18b)

$2x_3 + 2 - \lambda_3 - e_3 = 0$ (2.18c)

$x_1 - e_1' = 3$ (2.24a)

$x_2 - e_2' = 8$ (2.24b)

$$x_3 - e_3' = 2 \tag{2.24c}$$

2. Mengidentifikasi *complementary slackness*

   Berdasarkan (2.19) dan (2.24), (2.18) dan (2.22), dan Sifat 2.1, maka kondisi *complementary slackness* untuk persamaan (2.16) adalah:

   $\lambda_1 e_1' = 0$ $\qquad$ $e_1 x_1 = 0$

   $\lambda_2 e_2' = 0$ $\qquad$ $e_2 x_2 = 0$

   $\lambda_3 e_3' = 0$ $\qquad$ $e_3 x_3 = 0$

3. Menambahkan variabel buatan $\boldsymbol{a_i}$ untuk setiap kondisi *Khun Tucker* yang tidak memiliki variabel basis

   Persamaan (2.18) dan (2.24) tidak ada yang memiliki basis sehingga semuanya ditambahkan variabel buatan $\boldsymbol{a_i}$ sehingga bentuknya menjadi:

   $$4x_1 - \lambda_1 - e_1 + a_1 = 4 \tag{2.25a}$$

   $$4x_2 - \lambda_2 - e_2 + a_2 = 6 \tag{2.25b}$$

   $$-2x_3 + \lambda_3 + e_3 + a_3 = 2 \tag{2.25c}$$

   $$x_1 - e_1' + a_1' = 3 \tag{2.25d}$$

   $$x_2 - e_2' + a_2' = 8 \tag{2.25e}$$

   $$x_3 - e_3' + a_3' = 2 \tag{2.25f}$$

   Semua variabel non negatif.

4. Menentukan fungsi tujuan baru yang linear

   Bentuk fungsi linear baru yang linear untuk Contoh diatas adalah

Meminimumkan

$$w = a_1 + a_2 + a_3 + a_1' + a_2' + a_3' \quad (2.26)$$

Dengan kendala:

$$4x_1 - \lambda_1 - e_1 + a_1 = 4 \quad (2.25a)$$

$$4x_2 - \lambda_2 - e_2 + a_2 = 6 \quad (2.25b)$$

$$-2x_3 + \lambda_3 + e_3 + a_3 = 2 \quad (2.25c)$$

$$x_1 - e_1' + a_1' = 3 \quad (2.25d)$$

$$x_2 - e_2' + a_2' = 8 \quad (2.25e)$$

$$x_3 - e_3' + a_3' = 2 \quad (2.25f)$$

Semua variabel non negatif.

b. Penyelesaian model linear menggunakan Algoritma Genetika

Setelah mendapat fungsi (2.26) dan kendala (2.25) linear menggunakan metode *quadratic programming*, selanjutnya dilakukan penyelesaian model linear dengan algoritma genetika adalah sebagai berikut:

1. Generasi pertama

a. Membangkitkan populasi awal

Karena yang dicari adalah nilai $a_1, a_2, a_3, \ldots, e_3'$, maka variabel tersebut dijadikan sebagai gen-gen pembentuk kromosom , sehingga ditentukan gen yang digunakan sebanyak nilai yang akan dicari yaitu 18. Selanjutnya proses inisialisasi dilakukan dengan cara memberikan nilai awal gen-gen dengan nilai acak sesuai batasan yang telah ditentukan. Pada permasalahan ini jumlah populasi di tentukan sebanyak 18 sebagai berikut:

Kromosom (1) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [03; 06; 03; 05; 15; 01; 12; 18; 13; 09; 02; 05; 02; 09; 10; 15; 13; 07]$

Kromosom (2) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [04; 01; 13; 11; 15; 05; 16; 07; 09; 18; 11; 17; 13; 18; 02; 01; 08; 03]$

Kromosom (3) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [02; 17; 03; 18; 12; 09; 18; 01; 11; 14; 01; 07; 14; 04; 10; 03; 17; 13]$

Kromosom (4) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [09; 08; 12; 18; 05; 05; 16; 15; 18; 02; 08; 17; 13; 15; 06; 18; 04; 11]$

Kromosom (5) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [04; 05; 15; 06; 14; 01; 12; 11; 07; 08; 04; 03; 15; 02; 12; 01; 08; 01]$

Kromosom (6) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [09; 01; 08; 01; 02; 12; 05; 10; 10; 17; 10; 10; 16; 04; 05; 18; 09; 10]$

Kromosom (7) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [13; 17; 07; 04; 09; 02; 01; 13; 14; 01; 11; 13; 03; 03; 10; 05; 15; 06]$

Kromosom (8) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [18; 10; 02; 13; 01; 13; 13; 07; 09; 09; 06; 16; 05; 10; 12; 10; 09; 16]$

Kromosom (9) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [18; 18; 18; 06; 07; 05; 10; 07; 10; 11; 10; 14; 12; 13; 16; 10; 08; 06]$

Kromosom (10) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [02; 13; 11; 17; 01; 03; 15; 18; 03; 07; 12; 15; 18; 01; 17; 09; 18; 04]$

Kromosom (11) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [03; 16; 18; 06; 17; 16; 14; 17; 10; 02; 17; 10; 07; 16; 04; 17; 02; 06]$

Kromosom (12) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [12; 14; 14; 16; 08; 04; 18; 13; 05; 17; 07; 11; 11; 16; 12; 12; 07; 01]$

Kromosom (13) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [17; 02; 17; 03; 02; 07; 16; 07; 06; 04; 07; 17; 03; 02; 08; 05; 10; 18]$

Kromosom (14) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [09; 04; 01; 07; 05; 02; 10; 12; 13; 15; 06; 14; 07; 02; 05; 13; 14; 09]$

Kromosom (15) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [15; 02; 18; 06; 16; 08; 15; 04; 09; 07; 12; 08; 04; 01; 18; 11; 13; 09]$

Kromosom (16) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [04; 06; 13; 01; 01; 16; 02; 11; 08; 04; 13; 14; 05; 07; 15; 04; 12; 15]$

Kromosom (17) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2'; e_3']$

$= [04; 16; 17; 07; 14; 18; 17; 13; 03; 03; 08; 11; 01; 07; 04; 18; 15; 01]$

Kromosom (18) $= [a_1; a_2; a_3; a_1'; a_2'; a_3'; x_1; \lambda_1; e_1; x_2; \lambda_2; e_2; x_3; \lambda_3; e_3; e_1'; e_2';$

$e_3{}']$

$= [01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03; 11; 13; 03; 18]$

b. Seleksi

Permasalahan yang ingin diselesaikan adalah nilai variabel $a_1, a_2, a_3, \ldots, e_3{}'$ yang meminimumkan $a_1 + a_2 + a_3 + a_1{}' + a_2{}' + a_3{}'$, maka fungsi objektif yang dapat digunakan untuk mendapatkan solusi adalah fungsi objektif (kromosom) = $a_1 + a_2 + a_3 + a_1{}' + a_2{}' + a_3{}'$.

Perhitungan nilai fungsi objektif dari kromosom yang telah dibangkitkan adalah sebagai berikut:

f_{ob} (kromosom 1) = 33

f_{ob} (kromosom 2) = 49

f_{ob} (kromosom 3) = 61

f_{ob} (kromosom 4) = 57

f_{ob} (kromosom 5) = 45

f_{ob} (kromosom 6) = 33

f_{ob} (kromosom 7) = 52

f_{ob} (kromosom 8) = 57

f_{ob} (kromosom 9) = 72

f_{ob} (kromosom 10) = 47

f_{ob} (kromosom 11) = 76

f_{ob} (kromosom 12) = 68

f_{ob} (kromosom 13) = 48

f_{ob} (kromosom 14) = 28

f_{ob} (kromosom 15) = 65

f_{ob} (kromosom 16) = 41

f_{ob} (kromosom 17) = 76

f_{ob} (kromosom 18) = 43

Rata-rata dari fungsi objektif adalah : 52,83

Proses seleksi dilakukan dengan cara membuat kromosom yang mempunyai nilai fungsi objektif kecil mempunyai kemungkinan terpilih yang besar. Untuk itu dapat digunakan fungsi *fitness* sesuai dengan persamaan (2.13). Sehingga didapat nilai *fitness* sebagai berikut:

Fitness 1 = 0,0294

Fitness 2 = 0,0200

Fitness 3 = 0,0161

Fitness 4 = 0,0172

Fitness 5 = 0,0217

Fitness 6 = 0,0294

Fitness 7 = 0,0189

Fitness 8 = 0,0172

Fitness 9 = 0,0137

Fitness 10 = 0,0208

Fitness 11 = 0,0130

Fitness 12 = 0,0145

Fitness 13 = 0,0204

Fitness 14 = 0,0345

Fitness 15 = 0,0152

Fitness 16 = 0,0238

Fitness 17 = 0,0130

Fitness 18 = 0,0227

Total *fitness* = 0,3615

Tahap selanjutnya adalah mencari probabilitas terpilih. Dilakukan perhitungan menggunakan persamaan (2.14), dan didapat hasil sebagai berikut:

$$P(1) = \frac{0{,}0294}{0{,}3615} = 0{,}081$$

$$P(2) = \frac{0{,}0200}{0{,}3615} = 0{,}055$$

$$P(3) = \frac{0{,}0161}{0{,}3615} = 0{,}045$$

$$P(4) = \frac{0{,}0172}{0{,}3615} = 0{,}048$$

$$P(5) = \frac{0{,}0217}{0{,}3615} = 0{,}060$$

$$P(6) = \frac{0{,}0294}{0{,}3615} = 0{,}081$$

$$P(7) = \frac{0,0189}{0,3615} = 0,052$$

$$P(8) = \frac{0,0172}{0,3615} = 0,048$$

$$P(9) = \frac{0,0137}{0,3615} = 0,038$$

$$P(10) = \frac{0,0208}{0,3615} = 0,058$$

$$P(11) = \frac{0,0130}{0,3615} = 0,036$$

$$P(12) = \frac{0,0145}{0,3615} = 0,040$$

$$P(13) = \frac{0,0204}{0,3615} = 0,056$$

$$P(14) = \frac{0,0345}{0,3615} = 0,095$$

$$P(15) = \frac{0,0152}{0,3615} = 0,042$$

$$P(16) = \frac{0,0238}{0,3615} = 0,066$$

$$P(17) = \frac{0,0130}{0,3615} = 0,036$$

$$P(18) = \frac{0,0227}{0,3615} = 0,063$$

Berdasarkan hasil perhitungan diatas, didapatkan kromosom ke-14 yang mempunyai *fitness* paling besar maka kromosom tersebut mempunyai probabilitas untuk terpilih pada generasi selanjutnya lebih besar dari kromosom lainnya. Untuk proses seleksi digunakan *roulette wheel*, untuk itu dicari nilai kumulatif probabilitasnya menggunakan persamaan (2.15) sebagai berikut:

$C(1) = 0{,}081$

$C(2) = 0{,}136$

$C(3) = 0{,}181$

$C(4) = 0{,}229$

$C(5) = 0{,}289$

$C(6) = 0{,}370$

$C(7) = 0{,}422$

$C(8) = 0{,}470$

$C(9) = 0{,}508$

$C(10) = 0{,}566$

$C(11) = 0{,}602$

$C(12) = 0{,}642$

$C(13) = 0{,}698$

$C(14) = 0{,}793$

$C(15) = 0{,}835$

$C(16) = 0{,}901$

$C(17) = 0{,}937$

$C(18) = 1$

Setelah dihitung komulatif probabilitasnya maka proses seleksi menggunakan *roulette wheel* dapat dilakukan. Putar *roulette wheel* sebanyak jumlah populasi yaitu 18 kali (bangkitkan bilangan acak R antara 0 sampai 1 sebanyak 18) dan pada tiap putaran pilih satu kromosom untuk populasi baru. Misal didapatkan bilangan acak sebagai berikut:

$R(1) = 0{,}781$

$R(2) = 0{,}196$

$R(3) = 0{,}682$

$R(4) = 0{,}025$

$R(5) = 0{,}529$

$R(6) = 0{,}183$

$R(7) = 0{,}906$

$R(8) = 0{,}610$

$R(9) = 0{,}996$

$R(10) = 0{,}965$

$R(11) = 0{,}399$

$R(12) = 0{,}581$

$R(13) = 0{,}988$

$R(14) = 0{,}482$

$R(15) = 0{,}109$

$R(16) = 0{,}301$

$R(17) = 0{,}689$

$R(18) = 0{,}083$

Berdasarkan persamaan (2.16), kromosom baru yang menjadi induk adalah bilangan acak pertama R(1) lebih besar dari C(13) dan lebih kecil daripada C(14), maka pilih kromosom (14) sebagai kromosom pada populasi baru dari bilangan acak yang telah dibangkitkan diatas, maka populasi kromosom baru hasil proses seleksi adalah sebagai berikut:

Kromosom (1) = Kromosom (14)

Kromosom (2) = Kromosom (4)

Kromosom (3) = Kromosom (13)

Kromosom (4) = Kromosom (1)

Kromosom (5) = Kromosom (10)

Kromosom (6) = Kromosom (4)

Kromosom (7) = Kromosom (17)

Kromosom (8) = Kromosom (12)

Kromosom (9) = Kromosom (18)

Kromosom (10) = Kromosom (18)

Kromosom (11) = Kromosom (7)

Kromosom (12) = Kromosom (11)

Kromosom (13) = Kromosom (18)

Kromosom (14) = Kromosom (9)

Kromosom (15) = Kromosom (2)

Kromosom (16) = Kromosom (6)

Kromosom (17) = Kromosom (13)

Kromosom (18) = Kromosom (2)

Kromosom baru hasil proses seleksi adalah:

Kromosom (1) $= [09; 04; 01; 07; 05; 02; 10; 12; 13; 15; 06; 14; 07; 02; 05; 13; 14; 09]$

Kromosom (2) $= [09; 08; 12; 18; 05; 05; 16; 15; 18; 02; 08; 17; 13; 15; 06; 18; 04; 11]$

Kromosom (3) $= [17; 02; 17; 03; 02; 07; 16; 07; 06; 04; 07; 17; 03; 02; 08; 05; 10; 18]$

Kromosom (4) $= [03; 06; 03; 05; 15; 01; 12; 18; 13; 09; 02; 05; 02; 09; 10; 15; 13; 07]$

Kromosom (5) $= [02; 13; 11; 17; 01; 03; 15; 18; 03; 07; 12; 15; 18; 01; 17; 09; 18; 04]$

Kromosom (6) $= [09; 08; 12; 18; 05; 05; 16; 15; 18; 02; 08; 17; 13; 15; 06; 18; 04; 11]$

Kromosom (7) $= [04; 16; 17; 07; 14; 18; 17; 13; 03; 03; 08; 11; 01; 07; 04; 18; 15; 01]$

Kromosom (8) $= [12; 14; 14; 16; 08; 04; 18; 13; 05; 17; 07; 11; 11; 16; 12; 12; 07; 01]$

Kromosom (9) $= [01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03; 11;$

$13; 03; 18]$

Kromosom (10) $= [01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03; 11; 13; 03; 18]$

Kromosom (11) $= [13; 17; 07; 04; 09; 02; 01; 13; 14; 01; 11; 13; 03; 03; 10; 05; 15; 06]$

Kromosom (12) $= [03; 16; 18; 06; 17; 16; 14; 17; 10; 02; 17; 10; 07; 16; 04; 17; 02; 06]$

Kromosom (13) $= [01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03; 11; 13; 03; 18]$

Kromosom (14) $= [18; 18; 18; 06; 05; 02; 10; 07; 10; 11; 10; 14; 12; 13; 16; 10; 08; 06]$

Kromosom (15) $= [04; 01; 13; 11; 15; 05; 16; 07; 09; 18; 11; 17; 13; 18; 02; 01; 08; 03]$

Kromosom (16) $= [09; 01; 08; 01; 02; 12; 05; 10; 10; 17; 10; 10; 16; 04; 05; 18; 09;10]$

Kromosom (17) $= [17; 02; 17; 03; 02; 07; 16; 07; 06; 04; 07; 17; 03; 02; 08; 05; 10; 18]$

Kromosom (18) $= [04; 01; 13; 11; 15; 05; 16; 07; 09; 18; 11; 17; 13; 18; 02; 01; 08; 03]$

c. *Crossover*

Setelah proses seleksi maka proses selanjutnya adalah proses *crossover*. Metode yang digunakan salah satunya adalah *one-cut point*, yaitu memilih secara acak satu posisi dalam kromosom induk kemudian saling menukar gen. Kromosom yang dijadikan induk dipilih secara acak dan jumlah kromosom yang mengalami *crossover* dipengaruhi oleh *crossover probability* (ρ_c). Pada umumnya (ρ_c) ditentukan mendekati 1, misalnya 0,8 (Suyanto, 2005).

Misal ditentukan *crossover probability* adalah sebesar 0,8. Proses selanjutnya adalah dengan membangkitkan bilangan acak R antara 0 dan 1 sebanyak populasi yaitu 18 dan didapat bilangan acak sebagai berikut:

$R(1) = 0{,}805$

$R(2) = 0{,}078$

$R(3) = 0{,}738$

$R(4) = 0{,}432$

$R(5) = 0{,}058$

$R(6) = 0{,}967$

$R(7) = 0{,}458$

$R(8) = 0{,}199$

$R(9) = 0{,}082$

$R(10) = 0{,}122$

$R(11) = 0{,}931$

$R(12) = 0{,}466$

$R(13) = 0{,}288$

$R(14) = 0{,}027$

$R(15) = 0{,}882$

$R(16) = 0{,}591$

$R(17) = 0{,}987$

$R(18) = 0{,}902$

Kromosom ke-k akan dipilih sebagai induk jika $R(k) < \rho_c$, dari bilangan acak yang telah dibangkitkan. Sehingga kromosom yang menjadi induk adalah kromosom (2), kromosom (3), kromosom (4), kromosom (5), kromosom (7), kromosom (8), kromosom (9), kromosom (10), kromosom (12), kromosom (13), kromosom (14), dan kromosom (16).

Setelah melakukan pemilihan induk, proses selanjutnya adalah menentukan posisi Posisi *cut-point crossover*. Posisi *cut-point crossover* dipilih menggunakan bilangan acak 1 – (panjang kromosom – 1), dalam persoalan ini berarti dipilih bilangan acak 1-17 sebanyak jumlah *crossover* yang terjadi yaitu 12. Sehingga didapat posisi *cut-point crossover* sebagai berikut:

$C(1) = 3$

$C(2) = 4$

$C(3) = 7$

$C(4) = 14$

$C(5) = 15$

$C(6) = 7$

$C(7) = 8$

$C(8) = 11$

$C(9) = 4$

$C(10) = 11$

$C(11) = 5$

$C(12) = 12$

Setelah menentukan posisi *cut-point crossover*, selanjutnya akan dibentuk populasi kromosom setelah mengalami *crossover*. Dengan demikian populasi kromosom yang mengalami *crossover* adalah:

Offspring (1) diisi dengan cara melihat nilai C(1). Karena C(1) = 3, maka diambil 3 gen pertama dari kromosom (2), sedangkan gen selanjutnya diambil dari kromosom (3). Hal yang sama juga berlaku untuk *offspring* selanjutnya. Sehingga *offspring* yang terbentuk adalah

Offspring (1) = Kromosom (2) >< Kromosom (3)

$= [09; 08; 12; 03; 02; 07; 16; 07; 06; 04; 07; 17; 03; 02; 08; 05; 10; 18]$

Offspring (2) = Kromosom (3) >< Kromosom (4)

$= [17; 02; 17; 03; 15; 01; 12; 18; 13; 09; 02; 05; 02; 09; 10; 15; 13; 07]$

Offspring (3) = Kromosom (4) >< Kromosom (5)

$$= [03; 06; 03; 05; 15; 01; 12; 18; 03; 07; 12; 15; 18; 01; 17;$$
$$09; 18; 04]$$

Offspring (4) = Kromosom (5) >< Kromosom (7)

$$= [02; 13; 11; 17; 01; 03; 15; 18; 03; 07; 12; 15; 18; 01; 04;$$
$$18; 15; 01]$$

Offspring (5) = Kromosom (7) >< Kromosom (8)

$$= [04; 16; 17; 07; 14; 18; 17; 13; 03; 03; 08; 11; 01; 07; 04;$$
$$12; 07; 01]$$

Offspring (6) = Kromosom (8) >< Kromosom (9)

$$= [12; 14; 14; 16; 08; 04; 18; 02; 01; 05; 01; 08; 14; 03; 11;$$
$$13; 03; 18]$$

Offspring (7) = Kromosom (9) >< Kromosom (10)

$$= [01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03; 11;$$
$$13; 03; 18]$$

Offspring (8) = Kromosom (10) >< Kromosom (12)

$$= [01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 10; 07; 16; 04;$$
$$17; 02; 06]$$

Offspring (9) = Kromosom (12) >< Kromosom (13)

$$= [03; 16; 18; 06; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03; 11;$$
$$13; 03; 18]$$

Offspring (10) = Kromosom (13) >< Kromosom (14)

$= [01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 14; 12; 13; 16; 10; 08; 06]$

Offspring (11) = Kromosom (14) >< Kromosom (16)

$= [18; 18; 18; 06; 07; 12; 05; 10; 10; 17; 10; 10; 16; 04; 05; 18; 09; 10]$

Offspring (12) = Kromosom (16) >< Kromosom (2)

$= [09; 01; 08; 01; 02; 12; 05; 10; 10; 17; 10; 10; 13; 15; 06; 18; 04; 11]$

Hasil *crossover* menjadi kromosom baru, sedangkan kromosom yang tidak menggalami *crossover* tetap digunakan untuk membentuk populasi. Dengan demikian populasi kromosom setelah mengalami proses *crossover* adalah:

Kromosom (1) $= [09; 04; 01; 07; 05; 02; 10; 12; 13; 15; 06; 14; 07; 02; 05; 13; 14; 09]$

Kromosom (2) $= [09; 08; 12; 03; 02; 07; 16; 07; 06; 04; 07; 17; 03; 02; 08; 05; 10; 18]$

Kromosom (3) $= [17; 02; 17; 03; 15; 01; 12; 18; 13; 09; 02; 05; 02; 09; 10; 15; 13; 07]$

Kromosom (4) $= [03; 06; 03; 05; 15; 01; 12; 18; 03; 07; 12; 15; 18; 01; 17; 09; 18; 04]$

Kromosom (5) $= [02; 13; 11; 17; 01; 03; 15; 18; 03; 07; 12; 15; 18; 01; 04; 18; 15; 01]$

Kromosom (6) = $[09; 08; 12; 18; 05; 05; 16; 15; 18; 02; 08; 17; 13; 15; 06;$
$18; 04; 11]$

Kromosom (7) = $[04; 16; 17; 07; 14; 18; 17; 13; 03; 03; 08; 11; 01; 07; 04;$
$12; 07; 01]$

Kromosom (8) = $[12; 14; 14; 16; 08; 04; 18; 02; 01; 05; 01; 08; 14; 03; 11;$
$13; 03; 18]$

Kromosom (9) = $[01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03; 11;$
$13; 03; 18]$

Kromosom (10) = $[01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 10; 07; 16;$
$04; 17; 02; 06]$

Kromosom (11) = $[13; 17; 07; 04; 09; 02; 01; 13; 14; 01; 11; 13; 03; 03;$
$10; 05; 15; 06]$

Kromosom (12) = $[03; 16; 18; 06; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03;$
$11; 13; 03; 18]$

Kromosom (13) = $[01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 14; 12; 13;$
$16; 10; 08; 06]$

Kromosom (14) = $[18; 18; 18; 06; 07; 12; 05; 10; 10; 17; 10; 10; 16; 04;$
$05; 18; 09;10]$

Kromosom (15) = $[04; 01; 13; 11; 15; 05; 16; 07; 09; 18; 11; 17; 13; 18;$
$02; 01; 08; 03]$

Kromosom (16) = $[09; 01; 08; 01; 02; 12; 05; 10; 10; 17; 10; 10; 13; 15;$

$06; 18; 04; 11]$

Kromosom (17) $= [17; 02; 17; 03; 02; 07; 16; 07; 06; 04; 07; 17; 03; 02;$

$08; 05; 10; 18]$

Kromosom (18) $= [04; 01; 13; 11; 15; 05; 16; 07; 09; 18; 11; 17; 13; 18;$

$02; 01; 08; 03]$

d. Mutasi

Langkah pertama untuk menentukan kromosom yang mengalami mutasi adalah dengan menghitung panjang total gen yang ada dalam satu populasi. Dalam kasus ini panjang total gen sesuai dengan persamaan (2.15) adalah $18 \times 18 = 324$.

Untuk memilih posisi gen yang mengalami mutasi dilakukan dengan membangkitkan bilangan integer acak antara 1 sampai total gen, yaitu 1 sampai 324. Jika bilangan acak yang dibangkitkan lebih kecil daripada parameter *mutation probability* (ρ_m) maka pilih posisi tersebut sebagai sub-kromosom yang mengalami mutasi. Misal ρ_m kita tentukan 0,05 maka kemungkinan ada 5 % dari total gen yang mengalami mutasi. Sehingga didapat jumlah mutasi = $0{,}05 \times 324 = 16{,}2$ (dibulatkan menjadi 16).

Setelah membangkitkan bilangan acak antara 1 sampai 324 sebanyak 16 untuk memilih posisi gen yang mengalami mutasi dan membangkitkan bilangan acak 1 sampai 18 sebanyak 16 untuk menggantikan gen yang mengalami mutasi maka didapat bilangan acak sebagai berikut:

Tabel 2.2. Bilangan Acak untuk Mutasi Generasi Pertama

Posisi Gen	Bilangan Acak Pengganti
16	17
24	5
25	11
61	10
69	9
76	9
102	15
109	5
119	16
172	15
178	13
206	13
215	6
234	8
249	8
297	9

Dari tabel 2.2, diketahui bilangan acak pertama yang dibangkitkan terpilih posisi gen 16 yang mengalami mutasi. Dengan demikian yang akan mengalami mutasi adalah kromosom ke-1 gen nomor 16. Maka nilai gen pada posisi tersebut diganti dengan bilangan acak antara 1-18 yang sudah tercantum pada tabel 2.2 yaitu 17. Begitu seterusnya hingga bilangan acak terakhir yang telah dibangkitkan. Sehingga kromosom yang telah mengalami mutasi sebagai berikut:

Kromosom (1) $= [09; 04; 01; 07; 05; 02; 10; 12; 13; 15; 06; 14; 07; 02; 05;$
$\mathbf{17}; 14; 09]$

Kromosom (2) $= [09; 08; 12; 03; 02; \mathbf{05}; \mathbf{11}; 07; 06; 04; 07; 17; 03; 02; 08;$
$05; 10; 18]$

Kromosom (3) $= [17; 02; 17; 03; 15; 01; 12; 18; 13; 09; 02; 05; 02; 09; 10;$

$15; 13; 07]$

Kromosom (4) $= [03; 06; 03; 05; 15; 01; \mathbf{10}; 18; 03; 07; 12; 15; 18; 01; \mathbf{09}; 09; 18; 04]$

Kromosom (5) $= [02; 13; 11; \mathbf{09}; 01; 03; 15; 18; 03; 07; 12; 15; 18; 01; 04; 18; 15; 01]$

Kromosom (6) $= [09; 08; 12; 18; 05; 05; 16; 15; 18; 02; 08; \mathbf{15}; 13; 15; 06; 18; 04; 11]$

Kromosom (7) $= [\mathbf{05}; 16; 17; 07; 14; 18; 17; 13; 03; 03; \mathbf{16}; 11; 01; 07; 04; 12; 07; 01]$

Kromosom (8) $= [12; 14; 14; 16; 08; 04; 18; 02; 01; 05; 01; 08; 14; 03; 11; 13; 03; 18]$

Kromosom (9) $= [01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03; 11; 13; 03; 18]$

Kromosom (10) $= [01; 02; 15; 02; 16; 07; 09; 02; 01; \mathbf{15}; 01; 10; 07; 16; 04; \mathbf{13}; 02; 06]$

Kromosom (11) $= [13; 17; 07; 04; 09; 02; 01; 13; 14; 01; 11; 13; 03; 03; 10; 05; 15; 06]$

Kromosom (12) $= [03; 16; 18; 06; 16; 07; 09; \mathbf{13}; 01; 05; 01; 08; 14; 03; 11; 13; \mathbf{06}; 18]$

Kromosom (13) $= [01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 14; 12; 13; 16; 10; 08; \mathbf{08}]$

Kromosom (14) $= [18; 18; 18; 06; 07; 12; 05; 10; 10; 17; 10; 10; 16; 04;$

$\mathbf{08}; 18; 09;10]$

Kromosom (15) $= [04; 01; 13; 11; 15; 05; 16; 07; 09; 18; 11; 17; 13; 18;$

$02; 01; 08; 03]$

Kromosom (16) $= [09; 01; 08; 01; 02; 12; 05; 10; 10; 17; 10; 10; 13; 15;$

$06; 18; 04; 11]$

Kromosom (17) $= [17; 02; 17; 03; 02; 07; 16; 07; \mathbf{09}; 04; 07; 17; 03; 02;$

$08; 05; 10; 18]$

Kromosom (18) $= [04; 01; 13; 11; 15; 05; 16; 07; 09; 18; 11; 17; 13; 18;$

$02; 01; 08; 03]$

Setelah proses mutasi, maka telah diselesaikan satu iterasi dalam algoritma genetika atau disebut satu generasi dan didapatkan:

Kromosom (1) $= [09; 04; 01; 07; 05; 02; 10; 12; 13; 15; 06; 14; 07; 02; 05;$

$17; 14; 09]$

Kromosom (2) $= [09; 08; 12; 03; 02; 05; 11; 07; 06; 04; 07; 17; 03; 02; 08;$

$05; 10; 18]$

Kromosom (3) $= [17; 02; 17; 03; 15; 01; 12; 18; 13; 09; 02; 05; 02; 09; 10;$

$15; 13; 07]$

Kromosom (4) $= [03; 06; 03; 05; 15; 01; 10; 18; 03; 07; 12; 15; 18; 01; 09;$

$09; 18; 04]$

Kromosom (5) $= [02; 13; 11; 09; 01; 03; 15; 18; 03; 07; 12; 15; 18; 01; 04;$

$18; 15; 01]$

Kromosom (6) = $[09; 08; 12; 18; 05; 05; 16; 15; 18; 02; 08; 15; 13; 15; 06; 18; 04; 11]$

Kromosom (7) = $[05; 16; 17; 07; 14; 18; 17; 13; 03; 03; 16; 11; 01; 07; 04; 12; 07; 01]$

Kromosom (8) = $[12; 14; 14; 16; 08; 04; 18; 02; 01; 05; 01; 08; 14; 03; 11; 13; 03; 18]$

Kromosom (9) = $[01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 08; 14; 03; 11; 13; 03; 18]$

Kromosom (10) = $[01; 02; 15; 02; 16; 07; 09; 02; 01; 15; 01; 10; 07; 16; 04; 13; 02; 06]$

Kromosom (11) = $[13; 17; 07; 04; 09; 02; 01; 13; 14; 01; 11; 13; 03; 03; 10; 05; 15; 06]$

Kromosom (12) = $[03; 16; 18; 06; 16; 07; 09; 13; 01; 05; 01; 08; 14; 03; 11; 13; 06; 18]$

Kromosom (13) = $[01; 02; 15; 02; 16; 07; 09; 02; 01; 05; 01; 14; 12; 13; 16; 10; 08; 08]$

Kromosom (14) = $[18; 18; 18; 06; 07; 12; 05; 10; 10; 17; 10; 10; 16; 04; 08; 18; 09;10]$

Kromosom (15) = $[04; 01; 13; 11; 15; 05; 16; 07; 09; 18; 11; 17; 13; 18; 02; 01; 08; 03]$

$\text{Kromosom (16)} = [09; 01; 08; 01; 02; 12; 05; 10; 10; 17; 10; 10; 13; 15;$
$06; 18; 04; 11]$

$\text{Kromosom (17)} = [17; 02; 17; 03; 02; 07; 16; 07; 09; 04; 07; 17; 03; 02;$
$08; 05; 10; 18]$

$\text{Kromosom (18)} = [04; 01; 13; 11; 15; 05; 16; 07; 09; 18; 11; 17; 13; 18;$
$02; 01; 08; 03]$

e. Evaluasi

Selanjutnya akan diperiksa fungsi objektif setelah 1 generasi. Dan didapatkan hasil sebagai berikut:

f_{ob} (kromosom 1) = 28

f_{ob} (kromosom 2) = 39

f_{ob} (kromosom 3) = 55

f_{ob} (kromosom 4) = 33

f_{ob} (kromosom 5) = 39

f_{ob} (kromosom 6) = 57

f_{ob} (kromosom 7) = 77

f_{ob} (kromosom 8) = 68

f_{ob} (kromosom 9) = 43

f_{ob} (kromosom 10) = 43

f_{ob} (kromosom 11) = 52

f_{ob} (kromosom 12) = 66

f_{ob} (kromosom 13) = 43

f_{ob} (kromosom 14) = 79

f_{ob} (kromosom 15) = 49

f_{ob} (kromosom 16) = 33

f_{ob} (kromosom 17) = 48

f_{ob} (kromosom 18) = 49

Rata-rata dari fungsi objektif adalah : 50,06

Dapat dilihat dari hasil perhitungan fungsi objektif diatas bahwa setelah satu generasi, nilai hasil rata-rata fungsi objektif mengalami penurunan dibandingkan hasil rata-rata fungsi objektif pada saat sebelum mengalami seleksi *crossover* dan mutasi. Hal ini menunjukkan bahwa kromosom atau solusi yang dihasilkan setelah satu generasi lebih baik dibandingkan generasi sebelumya.

Selanjutnya generasi baru yang telah terbentuk akan mengalami proses yang sama seperti generasi sebelumya yaitu proses evaluasi, seleksi, *crossover*, dan mutasi yang kemudian akan menghasilkan kromosom-kromosom baru untuk generasi yang selanjutnya. Proses ini akan berulang sampai mendapatkan generasi terbaik atau akan berhenti setelah sejumlah generasi yang telah ditetapkan sebelumya.

Penyelesaian model linear dengan algoritma genetika selanjutnya dilakukan menggunakan *software* Matlab. Optimisasi Algoritma default dari Matlab merupakan optimisasi meminimumkan. Beberapa komponen Algoritma Genetika yang telah di set oleh Matlab yaitu jumlah populasi sebanyak 20, seleksi menggunakan *metode roulette*, *crossover probability* (ρ_c) = 0,8 dan *mutation probability* (ρ_c).

Hasil yang diperoleh dari perhitungan menggunakan *software* matlab (Lampiran 2) yaitu $x_1 = 3$, $x_2 = 8$, dan $x_3 = 2$. Sehingga nilai minimum untuk fungsi f nonlinear yaitu:

$$f(x) = 2{x_1}^2 + 2{x_2}^2 + {x_3}^2 - 4x_1 - 6x_2 + 2x_3$$

$$= (2)(3^2) + (2)(8^2) + 2^2 - (4)(3) - (6)(8) + (2)(2)$$

$$= 94$$